IN THE CLAIMS

1. (Currently Amended) A method comprising:

storing a plurality of partial keys corresponding to ~~a plurality~~ an equal number of original keys in memory, wherein storage of said plurality of partial keys requires less memory than storage of said equal number of original keys, and wherein said plurality of partial keys are used to determine hashing conflicts;

applying a hash function to an original key of said plurality equal number of original keys to generate a hash value; ~~wherein said hash function comprises any polynomial~~;

accessing the memory according to the hash value;

reading a stored partial key of said plurality of partial keys from the memory that corresponds to ~~the~~ said hash value, wherein said hash value is ~~based on said original key;~~ not stored in the memory; and

executing a conflict check by comparing a partial key derived from an incoming full key with the stored partial key ~~stored in the memory~~ where the partial key corresponds at most with one of the stored partial keys.

2. (Currently Amended) The method of Claim 1, wherein the partial key from the memory corresponding to the hash value includes saved bits comprising a consecutive, sequential string of bits~~, less than or equal to N, which is part~~ that is a subset of the original key where the subset includes a majority of bits of the original key.

3. (Currently Amended) The method of Claim 2, wherein the stored partial key ~~from the memory corresponding to the hash value~~ comprises a number of bits equal to or more than ~~the~~ a number of bits of the original key minus ~~the~~ a number of bits of the hash value.

4. (Original) The method of Claim 1, wherein the hash function is implemented by a linear feedback shift register.

5. (Currently amended) The method of Claim 1 further comprising applying a reverse function on the stored partial key ~~from the memory corresponding to the hash value~~ and the hash value to generate the original key.

6. (Original) The method of Claim 1 further comprising the steps of:

reading a result from the memory corresponding to the hash value; forwarding a packet of data according to the result read from the memory.

7. (Currently Amended) An apparatus comprising:

a memory which stores a plurality of partial keys used to determine hashing conflicts, wherein said plurality of partial keys correspond to a plurality of original keys, and wherein ~~storage~~ a data sum of said plurality of partial keys requires less memory than ~~storage~~ a data sum of said plurality of original keys;

a hash function block coupled to a memory that applies any polynomial to a full key and generates a partial key and a hash value which is used to point to one of the plurality of partial keys stored in the memory, wherein the plurality of partial keys include saved bits comprising ~~a~~ consecutive, sequential ~~string~~ strings of bits derived from the plurality of original ~~key~~ keys; and

a processor that compares one of the plurality of partial keys to the partial key comprising a majority of bits of the full key.

8. (Previously Presented) The apparatus of Claim 7, wherein the memory comprises a hash table size.

9. (Currently Amended) The apparatus of Claim 7, wherein the one of the plurality of partial keys stored in the memory comprises a number of bits equal to or more than ~~the~~ a number of bits of the ~~original~~ full key minus ~~the~~ a number of bits of the hash value.

10. (Previously Presented) The apparatus of Claim 7, wherein the hash function block comprises a linear feedback shift register.

11. (Previously Presented) The apparatus of Claim 9, wherein the linear feedback shift register corresponds to a Galois version.

12. (Previously Presented) The apparatus of Claim 9, wherein the linear feedback shift register corresponds to a Fibonacci version.

13. (Currently Amended) The apparatus of Claim 7 further including a reverse function generator coupled to the memory, wherein the reverse function generator ~~generates~~ restores the ~~original~~ full key based on the one of the plurality of partial keys stored in the memory and the hash value.

14. (Previously Presented) The apparatus of Claim 7 further comprising a forwarding engine coupled to the memory, wherein the forwarding engine forwards a data packet according to information read from the memory at an address corresponding to the one of the plurality of partial keys stored in the memory.

15. Cancelled

16. Cancelled

17. Cancelled

18. Cancelled

19. Cancelled

20. Canceled

21. Cancelled

22. Cancelled

23. Cancelled

24. Cancelled

25. Cancelled

26. Cancelled

27. (New) A method comprising:

generating a partial key and a hash value from an original key, where the partial key includes a consecutive subset of a majority of bits of the original key;

accessing a memory including multiple partial keys, where the hash value is not stored in the memory;

selecting a stored partial key from the memory that corresponds with the hash value;

comparing the partial key with the stored partial key; and

identifying a hash conflict when the partial key matches the stored partial key.

28. (New) The method of claim 27 where the multiple partial keys correspond to an equal number of multiple input keys and a data sum of all the multiple partial keys is less than a data sum of all the equal number of multiple input keys.

29. (New) The method of claim 28 where the multiple partial keys are each selectable according to a different hash value derived from one of the equal number of multiple input keys.

30. (New) The method of claim 27 where the comparing of the partial key includes reading less than the original key.

31. (New) The method of claim 32 where the hash value corresponds to a single entry in the memory.

32. (New) The method of claim 27 including recovering the original key by combining the stored partial key with the hash value.

33. (New) The method of claim 32 where the original key is recovered by a reverse linear feedback shift register.

34. (New) A system comprising:

a hash function configured to generate a hash value and a partial key from an input key, where the partial key includes a consecutive sequential string of bits derived from the input key;

a memory including stored partial keys that correspond to an equal number of input keys, where a stored partial key includes less data bits than its corresponding input key; and

a processor configured to:

identify the stored partial keys that is associated with the hash value;

compare the partial key to the stored partial key; and

identify a hash conflict when the partial key matches the stored partial key.

35. (New) The system of claim 34 where the stored partial key includes a majority of data bits of its corresponding input key.

36. (New) The system of claim 34 where the hash value is not stored in the memory.

37. (New) The system of claim 34 where the hash value includes an address location associated with the corresponding input key

38. (New) The system of claim 34 where the memory includes a data sum of the stored partial keys which is less than a data sum of the equal number of input keys.